Unsupervised / Supervised Classification

Triki Sadok

8 October 2019

Predicting player's positions by their scores from FIFA 2018

I-INTRODUCTION

Football is the most reputaded sport in the planet, followers are from all ages and sexes. Many people and start ups want to study the profiling of the players for some reason. To facilitate this task, we will classify players aith some critera to predict each player's position in the game.

1-Questions to answer:

-What is the criteria we will use to classify players? s'il est "Attacker", "Defender","goal keeper" ou "midfielder"

-how to classify a new player from it's caracteristics?

- will this classification be stable ?

2-Goals:

The goal is to find the best classes, or segments, where we will divide players into homogenious classes and then caracterise those classes.

II-Data manipulation:

```
base=read.csv('file.csv')
a2=base[,-c(1,2,3,4,8,9,10)]
head(base)
```

```
##   X     ID              name                club age height_cm weight_kg
## 1 1  20801 Cristiano Ronaldo      Real Madrid CF  32       185        80
## 2 2 158023          L. Messi        FC Barcelona  30       170        72
## 3 3 190871            Neymar Paris Saint-Germain  25       175        68
## 4 4 176580         L. Suárez        FC Barcelona  30       182        86
## 5 5 167495          M. Neuer    FC Bayern Munich  31       193        92
## 6 6 188545    R. Lewandowski    FC Bayern Munich  28       185        79
##   nationality eur_value eur_wage overall pac sho pas dri def phy
## 1    Portugal  95500000   565000      94  90  93  82  90  33  80
## 2   Argentina 105000000   565000      93  89  90  86  96  26  61
## 3      Brazil 123000000   280000      92  92  84  79  95  30  60
## 4     Uruguay  97000000   510000      92  82  90  79  87  42  81
## 5     Germany  61000000   230000      92  91  90  95  89  60  91
## 6      Poland  92000000   355000      91  81  88  75  86  38  82
##   international_reputation skill_moves weak_foot preferred_foot crossing
## 1                        5           5         4          Right       85
## 2                        5           4         4           Left       77
## 3                        5           5         5          Right       75
## 4                        5           4         4          Right       77
## 5                        5           1         4          Right       15
## 6                        4           3         4          Right       62
##   finishing heading_accuracy short_passing volleys dribbling curve
## 1        94              88            83      88        91    81
## 2        95              71            88      85        97    89
```

```
## 3        89               62            81      83        96    81
## 4        94               77            83      88        86    86
## 5        13               25            55      11        30    14
## 6        91               85            83      87        85    77
##   free_kick_accuracy long_passing ball_control acceleration sprint_speed
## 1                 76           77           93           89           91
## 2                 90           87           95           92           87
## 3                 84           75           95           94           90
## 4                 84           64           91           88           77
## 5                 11           59           48           58           61
## 6                 84           65           89           79           83
##   agility reactions balance shot_power jumping stamina strength long_shots
## 1      89        96      63         94      95      92       80         92
## 2      90        95      95         85      68      73       59         88
## 3      96        88      82         80      61      78       53         77
## 4      86        93      60         87      69      89       80         86
## 5      52        85      35         25      78      44       83         16
## 6      78        91      80         88      84      79       84         83
##   aggression interceptions positioning vision penalties composure marking
## 1         63            29          95     85        85        95      22
## 2         48            22          93     90        78        96      13
## 3         56            36          90     80        81        92      21
## 4         78            41          92     84        85        83      30
## 5         29            30          12     70        47        70      10
## 6         80            39          91     78        84        87      25
##   standing_tackle sliding_tackle gk_diving gk_handling gk_kicking
## 1              31             23         7          11         15
## 2              28             26         6          11         15
## 3              24             33         9           9         15
## 4              45             38        27          25         31
## 5              10             11        91          90         95
## 6              42             19        15           6         12
##   gk_positioning gk_reflexes
## 1             14          11
## 2             14           8
## 3             15          11
## 4             33          37
## 5             91          89
## 6              8          10
```

Ce jeu de données est celui du jeu Fifa 2017 contenant tous les joueurs inscrits dans l'organisme FIFPRO

Statistique Descreptive :

```
base::summary(a2[,1:10])
```

```
##       age          height_cm       weight_kg        overall
##  Min.   :16.00   Min.   :155.0   Min.   : 49.0   Min.   :46.00
##  1st Qu.:21.00   1st Qu.:177.0   1st Qu.: 70.0   1st Qu.:62.00
##  Median :25.00   Median :181.0   Median : 75.0   Median :66.00
##  Mean   :25.12   Mean   :181.3   Mean   : 75.4   Mean   :66.25
##  3rd Qu.:28.00   3rd Qu.:186.0   3rd Qu.: 80.0   3rd Qu.:71.00
##  Max.   :47.00   Max.   :205.0   Max.   :110.0   Max.   :94.00
##       pac             sho             pas             dri
##  Min.   :21.00   Min.   :14.00   Min.   :24.00   Min.   :24.00
##  1st Qu.:61.00   1st Qu.:44.00   1st Qu.:51.00   1st Qu.:57.00
```

```
##  Median :68.00   Median :56.00   Median :58.00   Median :64.00
##  Mean   :67.74   Mean   :53.49   Mean   :57.53   Mean   :62.59
##  3rd Qu.:75.00   3rd Qu.:64.00   3rd Qu.:65.00   3rd Qu.:70.00
##  Max.   :96.00   Max.   :93.00   Max.   :95.00   Max.   :96.00
##       def             phy
##  Min.   :12.0   Min.   :27.00
##  1st Qu.:34.0   1st Qu.:58.00
##  Median :52.0   Median :66.00
##  Mean   :49.4   Mean   :64.77
##  3rd Qu.:64.0   3rd Qu.:72.00
##  Max.   :90.0   Max.   :92.00
```

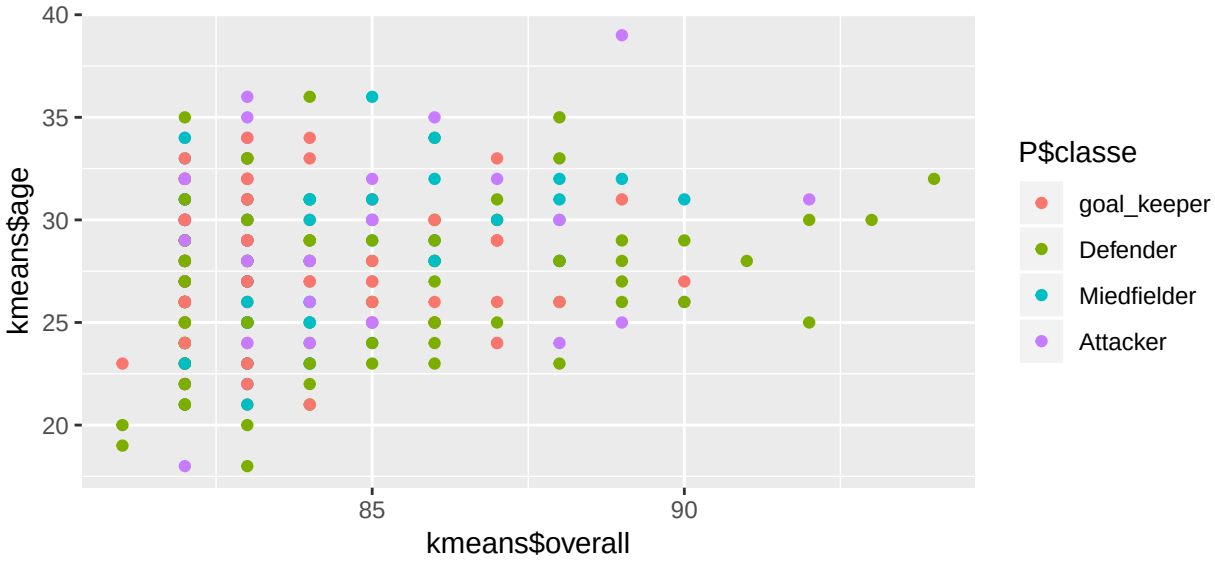
III-NON SUPERVISED CLASSIFICATION

1-K-means

```
##    base$name[1:300]    base$club[1:300]   classe
## 1 Cristiano Ronaldo      Real Madrid CF Defender
## 2           L. Messi        FC Barcelona Defender
## 3             Neymar Paris Saint-Germain Defender
## 4          L. Suárez        FC Barcelona Defender
## 5           M. Neuer    FC Bayern Munich Attacker
## 6     R. Lewandowski    FC Bayern Munich Defender
```

Results:

```
ggplot(kmeans, aes(kmeans$overall, kmeans$age, color = P$classe)) + geom_point()
```



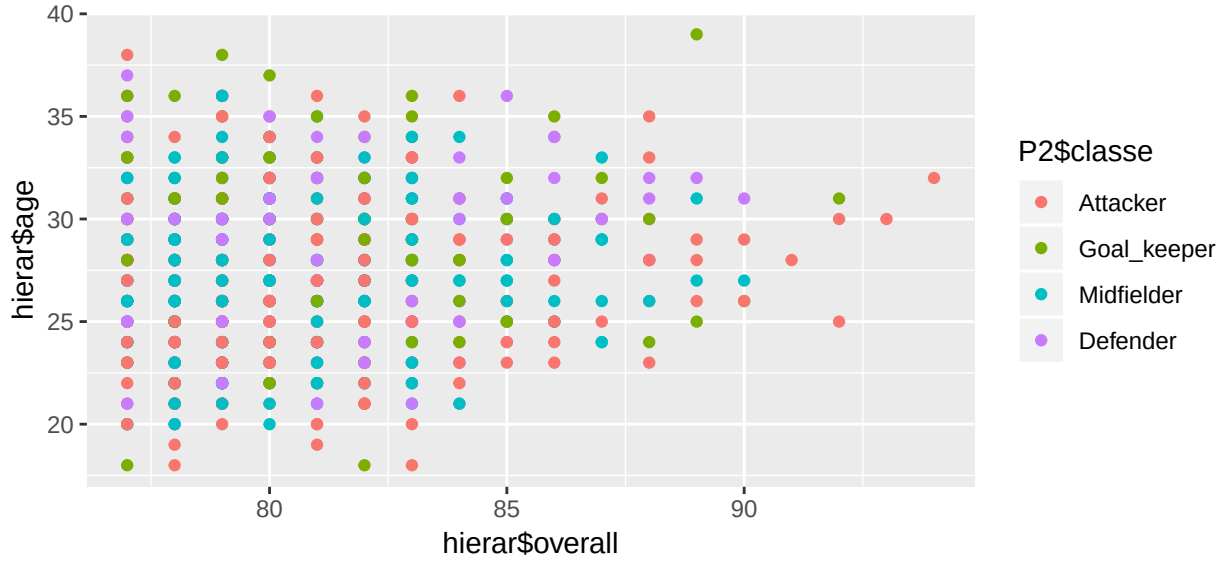
2-Hierarchical classification:

```
##    base$name[1:1000]    base$club[1:1000]       classe
```

```
## 1 Cristiano Ronaldo      Real Madrid CF    Attacker
## 2          L. Messi        FC Barcelona    Attacker
## 3            Neymar Paris Saint-Germain    Attacker
## 4         L. Suárez        FC Barcelona    Attacker
## 5          M. Neuer    FC Bayern Munich Goal_keeper
## 6    R. Lewandowski    FC Bayern Munich    Attacker
```

Representation :

```
ggplot(hierar, aes(hierar$overall, hierar$age, color =P2$classe )) + geom_point()
```



3 Final classification of players:

```
##    base$name[1:300]    base$club[1:300] Class Kmeans Class cluster
## 1 Cristiano Ronaldo      Real Madrid CF     Defender      Attacker
## 2          L. Messi        FC Barcelona     Defender      Attacker
## 3            Neymar Paris Saint-Germain     Defender      Attacker
## 4         L. Suárez        FC Barcelona     Defender      Attacker
## 5          M. Neuer    FC Bayern Munich     Attacker   Goal_keeper
## 6    R. Lewandowski    FC Bayern Munich     Defender      Attacker
```

-The best is to classify players to 4 classes.

-Those 4 classes will be:

- 1: **Attacker**
- 2: **Goal keeper**
- 3: **Midfielder**
- 4: **Defender**

IV-INDICE DE RAND ET STABILITE

1-Définition

"Rand" Index is a tool that can measure the **similarity** between two classifications. He is mostly used in automatic categorisation. It can mesure the consistancy between two possible classifications.

2- Algorithm(Simulation)

- Simulate a Data set X composed of multivariate normal distribution xi-> N(ui,sigma).

```
##         V1       V2
## 1 13.29271 29.33904
## 2 10.92474 28.36509
## 3 11.66589 29.21741
## 4 11.79364 30.52548
## 5 10.80924 29.70364
## 6 12.54325 31.30504
```

- Apply classification Algorithms A(k) et B(k) -> Pa(k) , Pb(k). (Here, we will use Kmeans A(k) et hierarchical classification B(K)).
- Study stability of partitions using Rand index and Rand index modified.
- sampling (stratified, SAS ..)Ej, applyed to Aj(k) et Bj(k) -> Pa(K) , Pb(K). Evaluate stability with IR , IRC
- Repeate N times the process.

```
RI <- function(D1,D2){

  library(fossil)

  v<-as.numeric(D1)
  vv<-as.numeric(D2)

   # rand index
  x <- abs(sapply(v, function(x) x - v))
  x[x > 1] <- 1
  y <- abs(sapply(vv, function(x) x - vv))
  y[y > 1] <- 1
  sg <- sum(abs(x - y))/2
  bc <- choose(dim(x)[1], 2)
  ri <- 1 - sg/bc

  # adj rand index
  a <- length(table(v))
  N <- length(v)
  ctab <- matrix(N, a, a)
  for (j in 1:a) {
    for (i in 1:a) {
      ctab[j, i] <- length(which(vv[which(v ==
                                            i)] == j))
    }
  }
```

```
  sumnij <- sum(choose(ctab, 2))
  sumai <- sum(choose(colSums(ctab), 2))
  sumbj <- sum(choose(rowSums(ctab), 2))
  Ntwo <- choose(N, 2)
  ari <- abs((sumnij - (sumai * sumbj)/Ntwo)/(0.5 * (sumai +
                                                     sumbj) - (sumai * sumbj)/Ntwo))

  #a=rand.index(v,vv)
  #b=adj.rand.index(v,vv)
  A=cbind(ri,ari)
  colnames(A)=c("rand_index","adj_rand_index")
  A=as.data.frame(A)
  #A=as.table(A)
  #METHOD <- "Augmented Dickey-Fuller Test"
  #names(ari) <- "rand index"
  #names(ri) <- "adjasted-rand index"
  #structure(list(statistic = ari, parameter = ri, method = METHOD, data.name ="A" ),
   #          class = "htest")
  return(A)
}
mean_rand11<-function(D,N,k,sampling){
  m=0
  p=0

  if (sampling=="stratification"){

    for(i in 1:N){
    kmeans=classifK_means(D,k)
    hierar=classif_hc(D,k,aggr[1],dd[1])
    strat4=stratif(hierar$a,kmeans$a)
    rand4=RI(strat$echantillon_1,strat$echantillon_2)
    m<-m+rand3$rand_index
    p<-p+rand3$adj_rand_index
    }}

  else if (sampling=="sample"){
    for(i in 1:N){

      D[sample(nrow(D), N), ]
      kmeans=classifK_means(D,k)
      hierar=classif_hc(D,k,aggr[1],dd[1])
      rand3=RI(kmeans$a,hierar$a)
      m<-m+rand3$rand_index
      p<-p+rand3$adj_rand_index
    }

  }

  mm<-m/N
  pp<-p/N
  AA=cbind(mm,pp)
  colnames(AA)=c("MEAN-RIx","MEAN-ARI")
  return(AA)
```

```
}
```

Using Rand index for FIFA18 dataset

```
mean=mean_rand11(a1,50,4,"sample")
```

```
## Loading required package: sp
## Loading required package: maps
##
## Attaching package: 'maps'
## The following object is masked from 'package:cluster':
##
##     votes.repub
## The following object is masked from 'package:plyr':
##
##     ozone
## Loading required package: shapefiles
## Loading required package: foreign
##
## Attaching package: 'shapefiles'
## The following objects are masked from 'package:foreign':
##
##     read.dbf, write.dbf
```

```
head(mean)
```

```
##       MEAN-RIx  MEAN-ARI
## [1,] 0.9118978 0.7865749
```

3-Comparing and Interpreting:

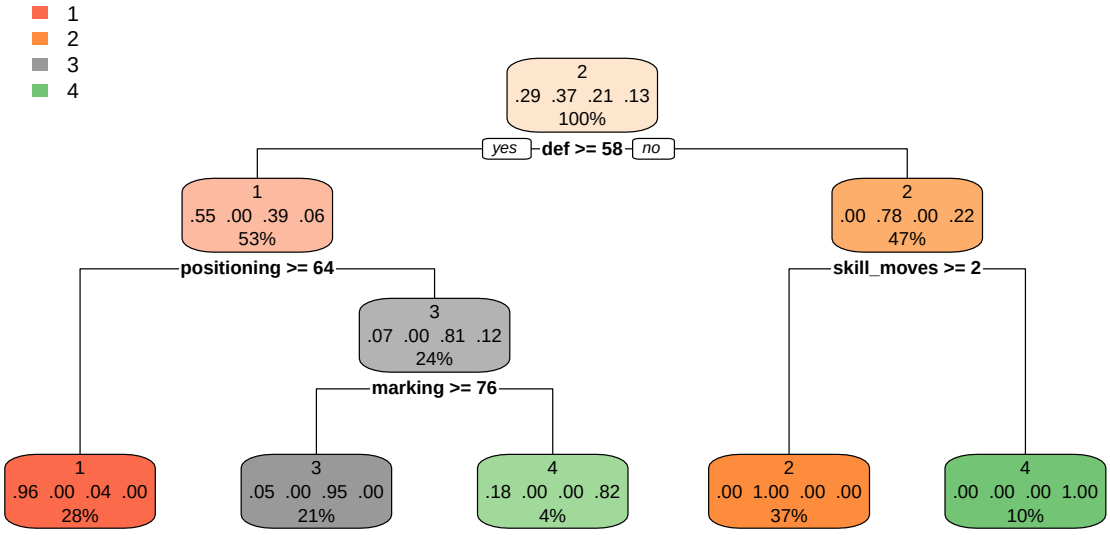
- *Simple sampling (SAS):*
- *Stratified Sampling:*

-We notice that the mean of Rand index (after 50 itérations) is close to 1 for the two sampling methods **simple** et **stratified**. That confirm that the two methods of classification used gives almost the same classes.

-Rand index modified gives same results (=0.7) and, it confirms the similarity of the methods. * We will the use then Kmeans and hierarchical classifications.

V-SUPERVISED CLASSIFICATION

1- Decision Tree:



VI-CONCLUSION:

-Finally, Thanks to a non supervised classification, with different methods, we have now a good partition of football international players : “Attacker”, “Defender”,“Goal keeper” et “Midfielder”.

-To verify the similarity in resulats given by Kmeans and hierarchical classification, Rand index is good, enough close to 1.

VII-ANNEXE

```
#INDICE DE RAND:
simul.X <- function(n1,n2,n3,n4){

  library(mvtnorm)
  mu=c(12,30)
  sigma=diag(1,2)
  x1=rmvnorm(n =n1, mean =mu, sigma)
  mu=c(25,36)
  sigma=diag(1,2)
  x2=rmvnorm(n = n2, mean =mu, sigma)
  mu=c(11,25)
  sigma=diag(1,2)
  x3=rmvnorm(n = n3, mean =mu, sigma)
  mu=c(5,35)
  sigma=diag(1,2)
  x4=rmvnorm(n = n4, mean =mu, sigma)
  X=rbind.data.frame(x1,x2,x3,x4)
```

```
return(X)
}

#HIERARCHICAL CLASSIFICATION

dd<-c("euclidian")
aggr<-c("ward","weighted","single","complete","average","flexible","gaverage")

classif_hc<-function(X,k,aggr,dd){

  library(cluster)
  classif<-agnes(X,method = aggr)
  cut<-cutree(classif,k=k)
  a=rep(0,nrow(X))
  X=cbind.data.frame(X,a)

  X[,48]=as.factor(cut)
  return(X)
}

#K-MEANS
classifK_means<-function(X,k){
  classif<-kmeans(X, k, iter.max = 10, nstart = 1)
  a=rep(0,nrow(X))
  X=cbind.data.frame(X,a)

  X[,48]=as.factor(classif$cluster)
  return(X)

}

 #INDICE DE RAND
RI <- function(D1,D2){

  library(fossil)

  v<-as.numeric(D1)
  vv<-as.numeric(D2)

   # rand index
  x <- abs(sapply(v, function(x) x - v))
  x[x > 1] <- 1
  y <- abs(sapply(vv, function(x) x - vv))
  y[y > 1] <- 1
  sg <- sum(abs(x - y))/2
  bc <- choose(dim(x)[1], 2)
  ri <- 1 - sg/bc

  # adj rand index
  a <- length(table(v))
  N <- length(v)
  ctab <- matrix(N, a, a)
  for (j in 1:a) {
```

```
    for (i in 1:a) {
      ctab[j, i] <- length(which(vv[which(v ==
                                          i)] == j))
    }
  }
  sumnij <- sum(choose(ctab, 2))
  sumai <- sum(choose(colSums(ctab), 2))
  sumbj <- sum(choose(rowSums(ctab), 2))
  Ntwo <- choose(N, 2)
  ari <- abs((sumnij - (sumai * sumbj)/Ntwo)/(0.5 * (sumai +
                                                     sumbj) - (sumai * sumbj)/Ntwo))

  #a=rand.index(v,vv)
  #b=adj.rand.index(v,vv)
  A=cbind(ri,ari)
  colnames(A)=c("rand_index","adj_rand_index")
  A=as.data.frame(A)
  #A=as.table(A)
  #METHOD <- "Augmented Dickey-Fuller Test"
  #names(ari) <- "rand index"
  #names(ri) <- "adjasted-rand index"
  #structure(list(statistic = ari, parameter = ri, method = METHOD, data.name ="A" ),
   #         class = "htest")
  return(A)
}

# STRATIFICATION
stratif <- function(D1,D2){
  library(splitstackshape)
  A=cbind.data.frame(D1,D2)
  c=stratified(A,c("D1","D2"),.2) #not sure about the size = 20%
  colnames(c)<-c("echantillon_1","echantillon_2")
  return(c)
}
```